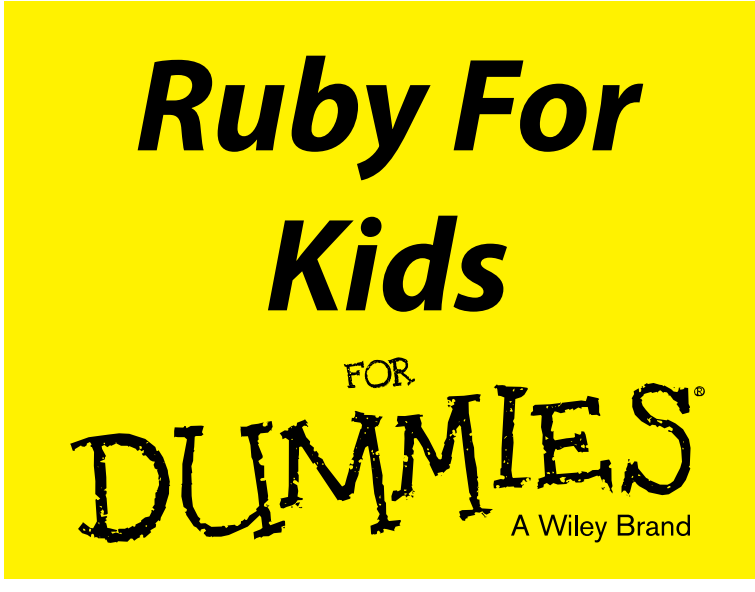
12 Fun Ruby Projects!

Ruby FOR KIDS

- Learn the basics of coding with Ruby
- Create your own games and apps
- Share your work with friends and family

Christopher Haupt
Mad Computer Scientist and Ruby Gemologist

FOR DUMMIES®
A Wiley Brand


Ruby For Kids
FOR
DUMMIES®
A Wiley Brand

Ruby For Kids

FOR DUMMIES®

A Wiley Brand

by Christopher Haupt

FOR DUMMIES®

A Wiley Brand

Ruby For Kids For Dummies®

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Library of Congress Control Number: 2015941961

ISBN 978-1-119-05590-7 (pbk); ISBN 978-1-119-05599-0 (ebk); ISBN 978-1-119-05600-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

Table of Contents

Introduction

Ruby For Kids For Dummies is an introduction to the basics of coding using the Ruby programming language. In each chapter, I walk you through a step-by-step set of instructions to create a Ruby program for your Mac or Windows computer. You don't need to have any programming experience to understand this book, but you do need to have a sense of curiosity and adventure!

The Ruby programming language has been around since the mid-1990s and has become very popular with web application programmers. It can be used for so much more than just web apps. In this book, you'll see that you can use Ruby for small "command line" tools and calculations; larger programs for home, work, or school; or even graphical games (and I'll show you a lot of games).

Ruby was designed by its creator Yukihiro Matsumoto to be both fun and productive. My hope is that as you explore the projects in this book, you'll definitely have fun and be inspired to continue to use Ruby (or any other programming language) to realize your own coding ideas.

Programming in general is similar to sports, music, or even creative arts. It's hard to just absorb a book on the subject and expect to understand it completely or start to gain mastery of the topic. Instead, you need to have keyboard time and practice. Even professional coders continue to practice throughout their careers.

By exploring and playing around with the projects here, you'll be taking the first steps down a really interesting Ruby-colored road.

About This Book

Programming is a large topic, and Ruby itself is a very powerful language. I'll be working to shed light on some of the more fundamental parts of Ruby and coding in general. There is no rush to finish the projects in the book. Go through each *Ruby For Kids For Dummies* project as quickly or slowly as you like. Each chapter's project is a self-contained useful utility or fun game. Along the way, you'll learn how to use the very same tools that the professionals use, and learn the kinds of techniques that will help you grow as a programmer.

You don't need to have any previous programming experience, but if you know a little, that's fine — you'll pick up how Ruby does things and also see some similarities to other languages. I'll show you the "Ruby way" when applicable, but I'll also show the easy way when you're just learning the concepts.

Topics covered in this book include the following:

- The general way to structure simple Ruby programs
- Ruby expressions and operators
- Organizing functionality using methods and objects
- Basic ways to represent data like numbers, strings, and arrays
- Using loops
- Making choices with `if...else` statements

Learning to program with Ruby isn't just about writing code in the language. You also need to learn about the tools, resources, and community that stand behind the language.

Ruby has become so popular because it's a relatively simple language to learn, and the tools needed to write Ruby, test it, and run

it are widely available and free. In this book, I help you get started with just a few basic, free, programs that do everything you need to create some pretty sophisticated pieces of software.

You'll also learn about general programming techniques, and most important, see a wide variety of projects that will pique your interest and hopefully encourage you to take your exploration to the next level.

To make this book easier to read, you'll want to keep in mind a few tips. First, all Ruby code and all terminal commands appear in monospaced type like this:

```
puts "hello programs! Welcome to Ruby"
```

The margins on a book page don't have the same room as your monitor likely does, so long lines of Ruby and any output it creates may break across multiple lines. Remember that your computer sees such lines as a single line of Ruby. I show that everything should be on one line by breaking it at a punctuation character or space and then indenting any overage, like so:

```
def room_type
["cave", "treasure room", "rock cavern", "tomb",
    "guard room", "lair"].sample
end
```

Ruby is case sensitive, which means that swapping the use of uppercase or lowercase letters or a combination of the two can break things. In order to make sure that you get the correct results from the projects in the book, always stick to the same capitalization and spelling that I use.

Ruby also cares about the kind of quotation marks that you use! So, if you see double quotes (") or single quotes ('), be sure to use what I show and make sure they're straight and not curly.

Foolish Assumptions

To understand programming, you need a bit of patience and the ability to think logically about a topic. You don't need to be a computer guru or a hacker. You don't need to be able to build a computer or take one apart (although that might be fun!). You don't need to know a bit from a byte or how many programmers it takes to screw in a new light bulb.

However, I do need to make some assumptions about you. I assume that you can turn your computer on, that you know how to use a mouse and a keyboard, and that you have a working Internet connection and web browser. You should also know how to find and launch programs on your computer.

In this book, I explain everything else you need to get set up and coding in Ruby.

Icons Used in This Book

Here's a list of the icons I use in this book to flag text and information that's especially noteworthy.

The Technical Stuff icon highlights technical details that you may or may not find interesting. Feel free to skip this information, but if you're the techie type, you might enjoy reading it.

The Tip icon highlights helpful tips that show you easy ways or shortcuts that will save you time or effort.

Whenever you see the Remember icon, pay close attention. You won't want to forget the information you're about to read — or, in some cases, I remind you about something that you've already learned that you may have forgotten.

Be careful. The Warning icon warns you of pitfalls to avoid.

Beyond the Book

I've put together a lot of extra content that you won't find in this book. Go online to find the following:

- **Cheat Sheet:** An online Cheat Sheet is available at `www.dummies.com/cheatsheet/rubyforkids`. Here, you find information on basic Ruby statements, conditions, loops and objects; a list of words that can't be used as Ruby variables or methods; a list of some of the useful methods provided by common Ruby classes; descriptions of common errors and what may cause them; and some small snippets of useful Ruby.

- **Web Extras:** Online articles covering additional topics are available at `www.dummies.com/extras/rubyforkids`. In these articles, I cover things like good ways to organize your Ruby class, some common Ruby shortcuts (also called "idiomatic Ruby"), Ruby troubleshooting tips, and more.

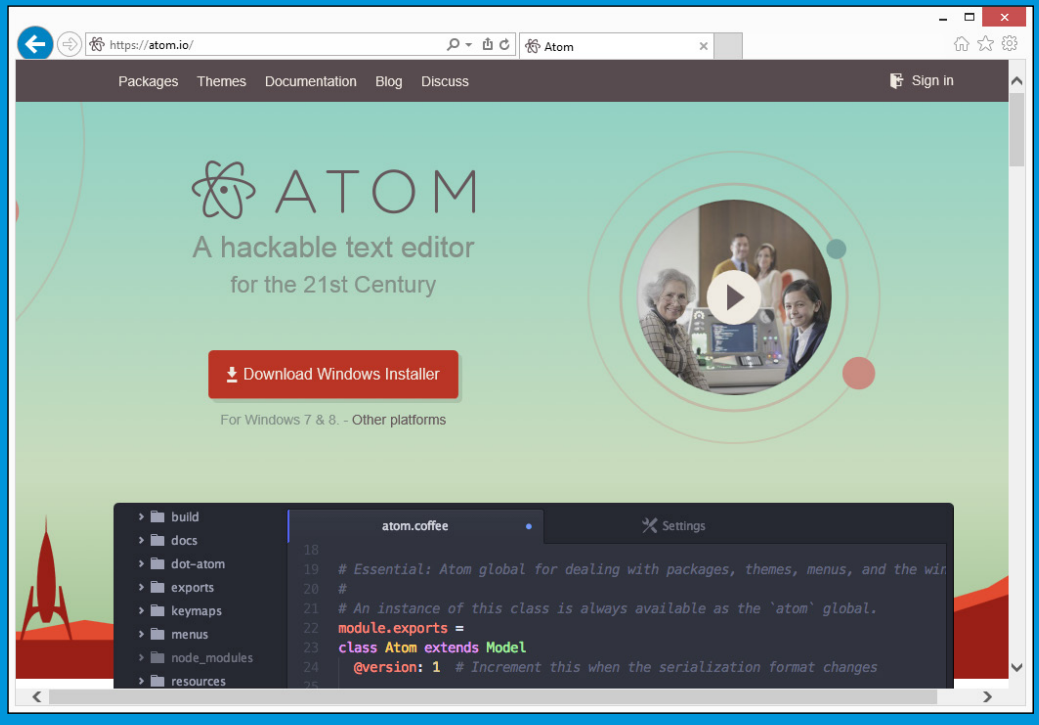
Where to Go from Here

Programming is a blast, and doubly awesome with Ruby. Even Ruby's creator wants you to have fun! After you learn the basics, you'll start to find all kinds of things you can do with your newfound powers.

I'm very interested to hear how it goes as you learn Ruby! If you want to show me your new ideas, bug fixes, or enhancements to my projects, or if you have programs you come up with on your own, you can do so on Facebook (`www.facebook.com/mobirobo`), on Twitter (`www.twitter.com/mobirobo_inc`), or via email at `ruby@mobirobo.com`.

Part I

The Most Basic Building Blocks



In this part . . .

For Dummies can help you get started with lots of subjects. Visit `www.dummies.com` to learn more and do more with *For Dummies!*

PROJECT

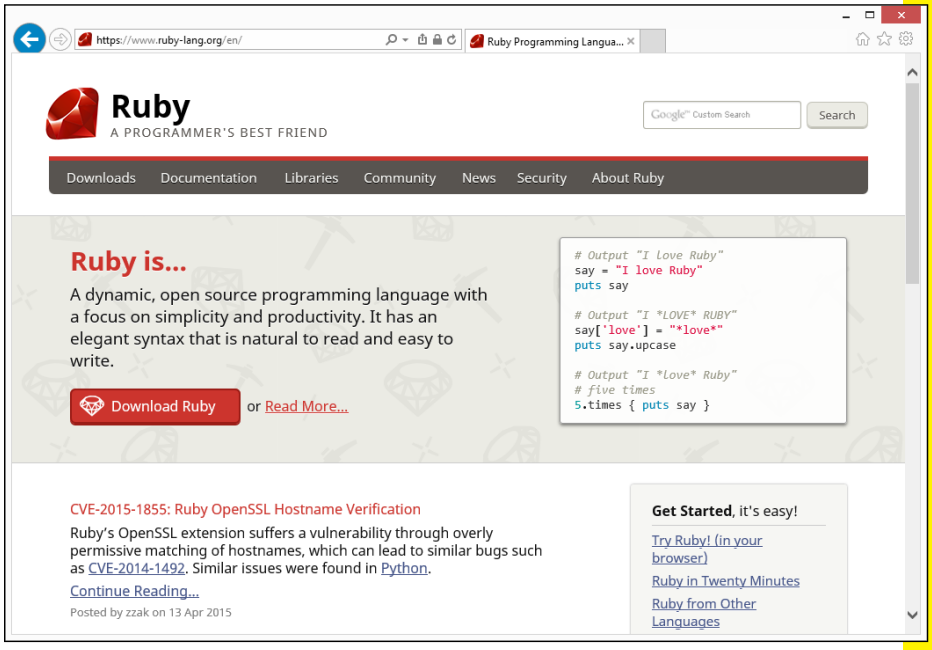
1

Getting Started with Ruby

Computers are almost everywhere today — from laptops, tablets, or phones, to TVs, watches, medical devices, kitchen appliances, cars, spaceships, big factories, little robots, and millions of other places large and small.

How do computers know what to do inside all these things? Someone has to teach them! Behind every cool animated movie, website, game, vehicle, or device, someone has worked hard to instruct a computer on how to perform its task. That person was a programmer.

In this chapter, I give you a little background about programming and how programmers organize their thoughts when writing computer software or code. I share some background about Ruby, the programming language I cover throughout this book. Then I tell you how to install the tools you'll use for all the projects in the rest of the book.



What Is Programming?

Computers are kind of dumb by themselves. Without a person to tell it exactly what to do, a computer will just sit there. Everything a computer does — and I mean everything, from the display of pictures and text on a screen, to the understanding of what you type on a keyboard or touch and swipe on a tablet — requires some software to interpret signals coming through the various circuits in one part of the computer and modify and send them to the right place in another part to get something done. That's a lot of work!

Fortunately, over the years, many smart people have come up with different ways to communicate clearly with computers. Writing instructions for a computer is called *programming* or *coding,* and the end result is a *program* or *software.*

A computer programming language shares many similarities to a human language. It has symbols and words (like nouns and verbs) that you put together following a *syntax* (rules for spelling, order, and punctuation).

When you start learning to program, you open up a wide world in which you can apply this knowledge when working with any technology that uses computers. You'll be able to read other people's programs to learn more about computers or to use code you write to solve homework problems, create puzzles, build a new game, create a website, or even control machines like robots.

Programs needs to be very precise in order to instruct a computer to do something. Imagine that you want to tell your friend to do something. For instance, how would you tell someone to sit down in a desk chair? You might say:

1. Pull the chair out.
2. Sit down.

Your friend is smart enough that your instructions make perfect sense, and she'll sit on the chair safely without falling over or anything crazy like that. People have a lot of knowledge they can use to interpret instructions like this.

Now, if you have to tell a computer to sit down, what would that be like? You have to be a *lot* more exact. For example, you would have to say:

1. Pull the chair away from the desk.
2. Walk around so your body is in front of the chair.
3. Turn around so your backside is facing the chair.
4. Make sure your body is exactly next to the chair.
5. Start bending your knees and lowering your body.
6. Keep bending your knees until your bottom makes contact with the seat of the chair.
7. Stop bending your knees when your weight is held by the chair.

Even these instructions might not be enough for a computer because they make some assumptions (like what your body parts are called).

Try it yourself: How would you tell a computer exactly how to do something like filling a glass with water?

Programmers need to think in this very detail-oriented way. As you learn to write computer programs, you'll get good at breaking a problem down into smaller and smaller parts. Each of those parts will eventually be a line of code that you create. Over time, you'll learn other techniques that help you identify the different objects you'll need to describe to the computer and the actions those objects will take. This will help you organize your code in

ways that make it possible to create very sophisticated software. Pretty cool, huh?

Why Ruby?

There are many different computer programming languages out there. Each language has strengths and weaknesses. Some languages are easier if you're trying to control large machines. Some languages are specialized for mobile apps — the kind on an iPhone, for example. Some languages make it easy to create websites. And some languages are for doing science and engineering.

A general-purpose programming language is good for many different kinds of projects. There are many general-purpose programming languages to choose from. The important thing when you're wanting to learn programming is to pick something and dive into training yourself to think like a programmer. When you learn one programming language, learning another one is much, much easier.

In this book, I use the language Ruby. Ruby is a flexible, general-purpose language that is useful for many kinds of projects. It was created in the mid-1990s in Japan by Yukihiro Matsumoto (best known by his nickname, "Matz"). Don't worry — you don't have to learn Japanese to program with Ruby! Today Ruby is used around the world for all kinds of projects, by beginners and professionals alike.

Matz had a wonderful philosophy in mind when creating Ruby: He wanted programmers to be productive, enjoy programming, and be happy. This is one of my favorite things about Ruby: As you learn it and write programs, you'll have fun!

What Tools Do You Need?

Most obviously, you need a computer that's running a current version of a consumer desktop operating system (Mac OS X or Windows).

If you're using a computer with Linux on it, you can still follow along with the projects in this book. I won't be going through the instructions here. Instead, check out the official Ruby documentation: `www.ruby-lang.org/en/documentation/installation`. As long as your selected approach installs at least version 1.9.3 of Ruby, you should be okay.

For the projects in this book, you need only a few basic tools, and they're all free.

First, you need Ruby installed, as well as some other software that helps Ruby use the capabilities of your computer. I walk you through how to install Ruby in this section.

Second, you need a text editor that is specifically for coding. Word processors don't work well when coding, so you'll use a tool that is built for programmers. There are a number of good, free code editors out there, and I help you install one of them in this section. (You may use any other editing program you like as long as it's a code editor of some kind.)

If you're on Windows

To run Ruby on Windows, you have to install Ruby and several developer tools. The following instructions have been tested with Windows 8 and 8.1.

1. Go to `http://rubyinstaller.org` in your web browser.
2. Click the big red Download button.

   A list of RubyInstallers appears.
3. Click Ruby 2.2.2 near the top of the RubyInstallers list (see Figure 1-1).

   Do *not* click Ruby 2.2.2 (x64).

   An installer program downloads to your computer.

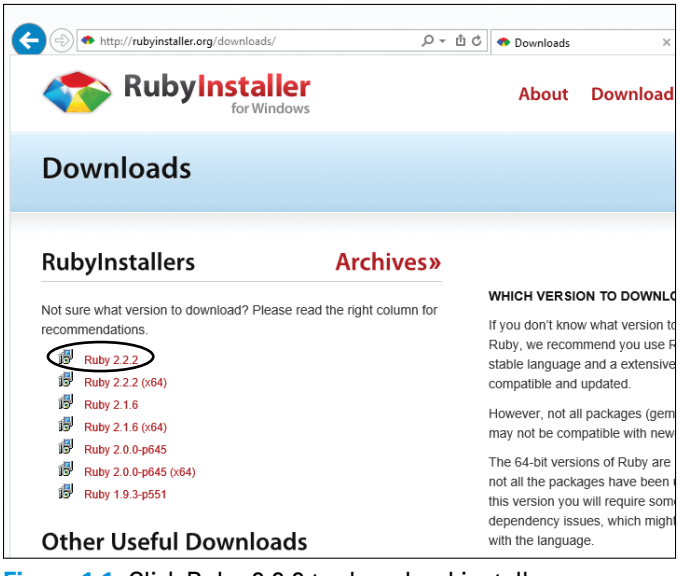


Figure 1-1: Click Ruby 2.2.2 to download installer.

4. Run the installer program by choosing Run Program (if Windows presents this option) or double-clicking the file when it's done downloading.

   The installer will ask you to select a language to use during installation. Accept the license, and then the installer will have you set some configuration options. Leave the default folder choice alone, but *uncheck* the Install Tcl/Tk Support check box (you won't be using it for this book), and make sure that the other two check boxes — Add Ruby Executables to Your PATH and Associate .rb and .rbw Files with This Ruby Installation — are selected (see Figure 1-2).

   When the installer is done, it will have created a topmost folder with all the Ruby software on your `C:` drive called `C:\Ruby22`. You can use Windows 8 Desktop and the File Explorer to confirm that it's there (as shown in Figure 1-3).

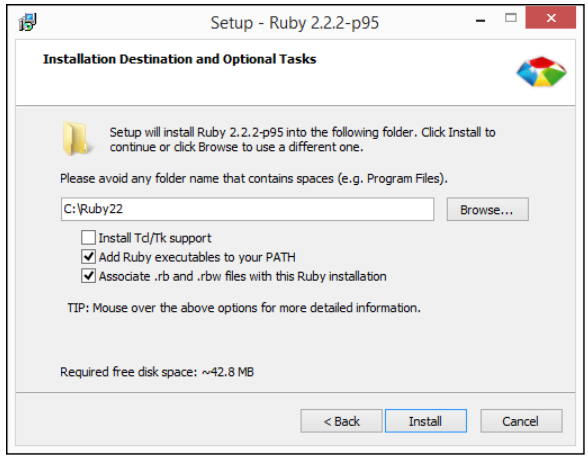


Figure 1-2: Setup Ruby installation settings.

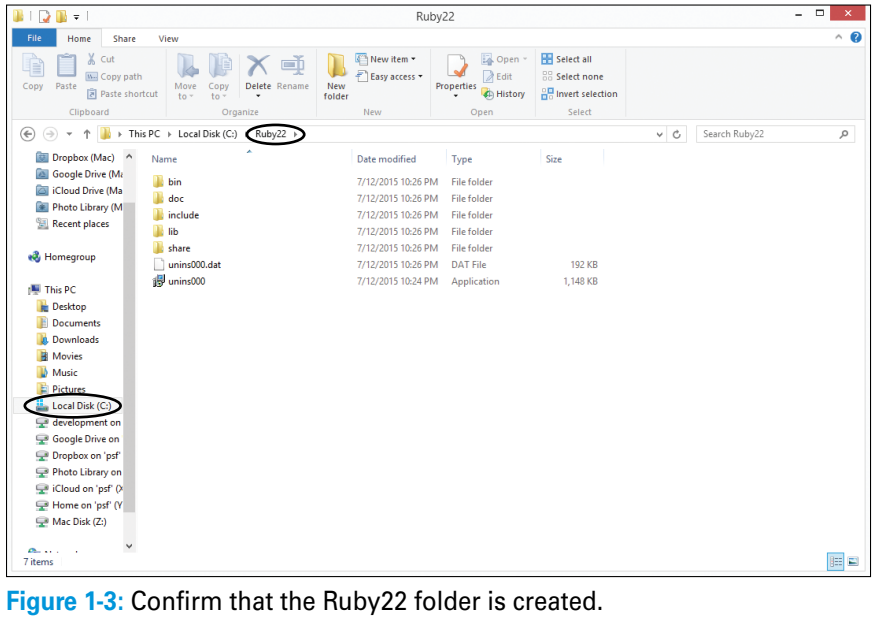


Figure 1-3: Confirm that the Ruby22 folder is created.

You must also download the Development Kit from `http://rubyinstaller.org` to get some of the cool tools used by the projects in this book. Follow these steps:

1. Go to the `http://rubyinstaller.org` in your web browser.

2. Scroll down to the Development Kit section and click the file under "For use with Ruby 2.0 and above (32bits version only)" (see Figure 1-4).

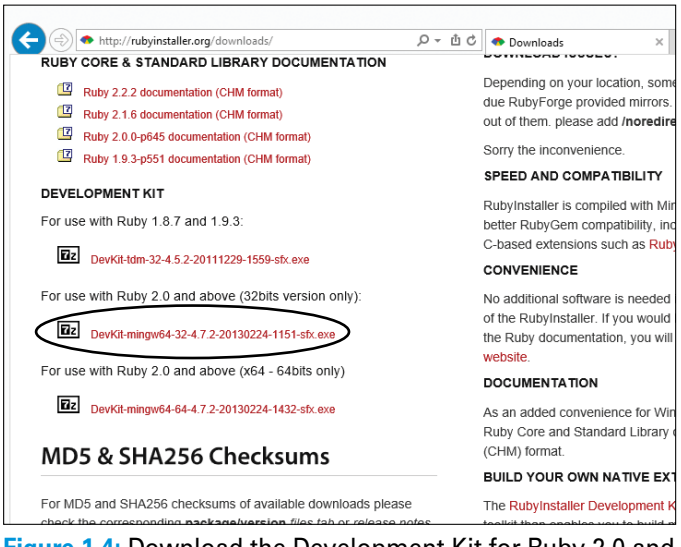


Figure 1-4: Download the Development Kit for Ruby 2.0 and above. Be sure to click the 32bits version.

An installer program downloads to your computer.

3. Run the Development Kit installer by choosing Run Program (if Windows presents this option) or double-clicking the file when it finishes downloading.